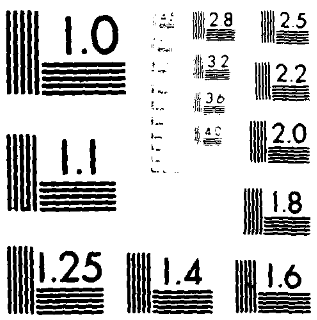

MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS 1963-A

June 1981 Report. No. STAN-CS-81-858

7

AD A112973

The Bit Operation Complexity of Approximate Evaluation of Matrix and Polynomial Products Using Modular Arithmetic

by

V. Pan

Research sponsored by

Office of Naval Research
National Science Foundation

Department of Computer Science

Stanford University
Stanford, CA 94305

The Bit-Operation Complexity of Approximate Evaluation of Matrix and Polynomial Products Using Modular Arithmetic

V. Pan*
Computer Science Department
Stanford University
Stanford, California 94305

Abstract. The approximate evaluation with a given precision of matrix and polynomial products is performed using modular arithmetic. The resulting algorithms are numerically stable. At the same time they are as fast as or faster than the algorithms with arithmetic operations over real or complex numbers.

* This research has been supported by National Science Foundation grant MCS-77-23738 and Office of Naval Research contract N00014-81-K-0269. Reproduction in whole or in part is permitted for any purpose of the United States government.

1. Introduction.

This paper continues our study (see [1,2]) of the bit-operation complexity of multiplication of two $n \times n$ matrices ($MM(n)$) and of two n-th degree polynomials ($PM(n)$) (the latter problem is also called convolution of vectors). We scale the binary approximations to the inputs and outputs of the problem to turn them into integers and then perform the computations over integers modulo K where K is appropriately chosen so that the desired approximate values of outputs are obtained involving fewer bit-operations. The same approach can be used for DFT problems and for any linear or bilinear (and even for any division free) arithmetic computational problems.

This idea is very simple but the results of our analysis seem meaningful for three reasons.

The first reason is that our estimates show that practically all fast algorithms for $MM(n)$ can be stabilized (if they are unstable) with no sacrifice in their rapidity. We hope that this our result will stop the discussion on the instability of fast matrix multiplication. Such a discussion recently has been renewed in the literature (cf. [3,4]) although with no substantial progress comparing, say, with the results of Ref. [5]. As a matter of fact, some ways of the stabilization of asymptotically fast algorithms for $MM(n)$ have already been described in [6, sections 1, 3, 16] but our present approach is more efficient. It successfully works even for problems of $MM(n)$ where n is moderate or small although in this paper we focus on the asymptotic estimates for the bit-operation complexity of algorithms where $n \to \infty$.

Secondly, our upper estimates for the bit-operation complexity of $PM(n)$ obtained here and in [1] differ from the lower estimates by the factor $\log^2 n$ rather than $\log n$ (in the cases of computation over complex numbers and over integers modulo K). It would be highly interesting for the theory of computation to reduce this gap, say, to $\log n$. This might be an easier problem than the reduction of the gap, $\log n$, between the lower and upper bounds on the number of arithmetic operations for the same problem, $PM(n)$.

Thirdly, we prove that, as could be expected, in the case of real inputs and constants the transition to computation modulo K enables us to accelerate the evaluation-interpolation algorithms for $PM(n)$ roughly in n times (cf. [1]).

In the next section we outline our approach and give preliminary estimates for a general class of bilinear computational problems. This class includes the cases of $MM(n)$ and $PM(n)$ which are further studied in the concluding Section 3.

We will use the following notation.

Notation. $T(MM(n))$, $T(PM(n))$, $t_a(p)$, $t_m(p)$ are the four minimum numbers of bit-operations required in order to solve the problems $MM(n)$, $PM(n)$, to add/subtract and to multiply two p-bit binary integers respectively. (Here and hereafter we assume that the moduli of inputs of $MM(n)$ and $PM(n)$ are bounded by a given constant $M > 0$ and that the solutions (outputs) are to be found with the errors less than a given c[illegible] $E > 0$.) $g(x) = O(f(x))$ if $|g(x)| < c|f(x)|$ for $|x| > c$ where $c > 0$ is a constant.

Hereafter all logarithms are to the base 2, the cases of real and complex inputs and outputs are studied simultaneously and the resulting asymptotic estimates hold in both cases.

2. The general approach and preliminary estimates.

Let $|x_i|$, $|y_j|$, the moduli of inputs of a given bilinear computational problem, be bounded by M. Let the outputs,

(2.1) $$w_k = \sum_{s=1}^{n} x_{g(s)} y_{h(s)} \delta(s,k) \qquad \text{where } \delta(s,k) \text{ is } 0 \text{ or } 1,$$

are to be evaluated with the absolute errors at most E, that is

(2.2) $$|w_k^* - w_k| < E \qquad \text{for all } k.$$

(The search of the approximate solutions to $PM(n)$ and $MM(n)$ is included here.) The bound (2.2) is guaranteed if (cf. (2.1))

(2.3) $$w_k^* = \sum_{s=1}^{n} x^*_{l(s)} y^*_{l(s)} \delta(s,k)$$

and if $x^*_{l(s)}$, $y^*_{l(s)}$ are b-bit binary approximations to x_i, y_j such that

(2.4) $$\forall i\colon |x_i^* - x_i| < e, \quad |x_i^*| \le |x_i|; \quad \forall j\colon |y_j^* - y_j| < e, \quad |y_j^*| \le |y_j|;$$

(2.5) $$E = 2Mne.$$

(To simplify our notation, we assume hereafter that

(2.6) $$e = E/2Mn = 2^{-h}, \quad h \text{ is an integer.})$$

Let b be the minimum integer under the above conditions, $b = O(h)$, and let

(2.7) $$\forall i\colon \tilde{x}_i = x^*/e, \quad \forall j\colon \tilde{y}_j = y_j/e, \quad \forall k\colon w_k^* = \tilde{w}_k e^2.$$

Then (cf. (2.3)-(2.7)) $\tilde{x}_i$, $\tilde{y}_j$, $\tilde{w}_k$ are binary integers and

(2.8) $$\forall i\colon |\tilde{x}_i| < M/e, \quad \forall j\colon |\tilde{y}_j| < M/e, \quad \forall k\colon |\tilde{w}_k| < (M/e)^2 n.$$

Let A be an algorithm with the inputs $\tilde{x}_i$, $\tilde{y}_j$ that evaluates the $\tilde{w}_k$ and uses only arithmetic operations over (Gaussian) integers. Then we can also assume that the operations of A are performed modulo K. If

(2.9) $$K \ge K_0 = (M/e)^2 n = 4M^4 n^3/E^2$$

we still obtain the same values $\tilde{w}_k$, (cf. (2.8)). Then it remains to shift the radix point of $\tilde{w}_k$ $2h$ bits left in order to obtain the desired w_k^*, (cf. (2.6)-(2.7)).

Remark. The main advantage of such an approach is that the approximations with the required precision are evaluated involving only (Gaussian) integers modulo K. This guarantees numerical stability of the algorithm assuming that K is not very large.

In the next two sections we apply this approach to obtain upper estimates for $T(MM(n))$ and $T(PM(n))$ using the following auxiliary upper estimates:

(2.10) $$t_a(p) = O(p), \quad t_m(p) = O(p \log p \log\log p),$$

(cf. [7-10]).

3. Applications to matrix and polynomial multiplication.

Let an algorithm, A, for $MM(n)$ use $O(n^q)$ arithmetic operations. We can always assume that no divisions are involved, (cf. [8] pp. 35–38). If all constants of A are integers we can choose any K that satisfies (2.9). If all constants of A are rational we can reduce the computation to the evaluation of the $s\tilde{w}_k$ with only integer constants. Here the integer s is "the accumulated denominator" of noninteger constants of A. We can assume (see [2] section 6) that

$$\log s = O(\log n). \tag{3.1}$$

In this case the computations modulo K give the values $\tilde{w}_k$ if K satisfies (2.9) and if K and s are relatively prime. (Then we add n^2 multiplications, $\tilde{w}_k = t(s\tilde{w}_k) \bmod K$ where $ts = 1 \bmod K$. Their cost is negligible because always $q \geq 2$, see [7–10].) We can assume that for some K, such that

$$K = O(K_0), \tag{3.2}$$

both conditions are satisfied. (Recall (3.1) to see that s can not be too large.)

It is known that the exponent q of $MM(n)$ can be chosen smaller than 2.496. (See [6,11], we cite [6] where "only" $q < 2.52$ is obtained because the origin of the basic design of [7] is explained in detail in the earlier paper [6] while in the otherwise successful paper [7] the confusing name, "Schönhage's construction", does not help the reader.) We can always assume that the constants of an algorithm for MM are algebraic numbers obtained from the solution of a system of polynomial equations.

We have come to the following upper estimate. (See (2.10), (3.2). For the transition from algebraic constants to rational ones see, for instance, [6, section 3].)

Theorem 1. *If there exists an algorithm for $MM(n)$ that uses $O(n^q)$ arithmetic operations then*

$$T(MM(n)) = O(n^q t_m(\log K_0)) \tag{3.3}$$

where $t_m(p)$ and K_0 satisfy (2.9), (2.10), $q < 2.496$.

In order to make the described above approach applicable to $PM(n)$, it is sufficient (cf. [8, pp. 86–87] or [9, p. 440]) to choose a prime K that satisfies (2.9) and such that

$$K = c \cdot 2^r + 1, \quad 2^r > 2n. \tag{3.4}$$

Let K be such a prime and let

$$K = O(K_0 n^\alpha), \quad (\alpha \geq 0 \text{ is a constant}). \tag{3.5}$$

(For the justification of (3.5) see [8, pp. 86–87] or [9, p. 440].)

Then $PM(n)$ can be solved by an evaluation-interpolation algorithm with $DFT(2^r)$ using $O(r2^r)$ additions, $O(2^r)$ multiplications modulo K, see [12]. This gives the following upper estimate, (cf. (2.10)).

Theorem 2.

$$T(PM(n)) = O(n \log n \log(K_0 n^{\alpha}) + n t_m(\log(K_0 n^{\alpha}))) \tag{3.6}$$

where K_0 satisfies (2.9), $\alpha \geq 0$ is a constant.

Corollary. *(cf. (2.10), (3.3)–(3.6).)*

$$T(MM(n)) = O(n^{q}), \quad T(PM(n)) = O(n \log^2 n) \quad \text{if } K_0 \text{ is a constant.}$$
$$T(MM(n)) = O(n^{q} \log n \log\log n \log\log\log n), \quad T(PM(n)) = O(n \log^2 n)$$

if $K_0 = O(n^{u})$, $u > 0$ is a constant.

$$T(MM(n)) = O(n^{q+v} \log n \log\log n), \quad T(PM(n)) = O(n^{1+v} \log n \log\log n)$$

if $\log K_0 = n^{v}$, *$v > 0$ is a constant.*

(3.3), (3.6) can be compared with the estimates for the bit-complexity of the same algorithms where the computations are in the fields of real or complex numbers, see [1,2]. In the complex case for $MM(n)$ and $PM(n)$ and in the real case for $MM(n)$ we have just obtained practically the same asymptotic upper estimates as in [1,2], up to a factor n^{ϵ} for $MM(n)$ where $\epsilon > 0$ is arbitrarily small. (See however our Remark above in Section 2.) The informational lower bound of Theorem 5.1 from [1] can be applied to the case of the evaluation modulo K also. Comparing with the upper bounds of Theorem 5.1 from [1] and of Theorem 2 above we notice that the gap between the lower and upper bounds on $T(PM(n))$ is $\log^2 n$ rather than $\log n$ even if $M^2 n/E$ is a constant. (Is the upper estimate sharp up to a constant factor?) In the real case the upper estimate (3.6) and the lower estimate of Theorem 6.1 from [1] show that the transition to the computations modulo K enables us to reduce $T(PM(n))$ roughly by the factor n.

Conclusion.

It is interesting to apply the above approach to the bilinear computational problems studied in [12]. Those are convolution of vectors and some problems that are encountered in signal processing. Then the resulting algorithms are faster than ones from [12], better structured and have guaranteed numerical stability. The only price for that is the use of modular arithmetic. In which cases is such a price too high?

References

1. V. Pan, "The bit-complexity of arithmetic algorithms," to appear in *Journal of Algorithms*.
2. V. Pan, "The bit-operation complexity of matrix multiplication and of all pair shortest path problem," to appear in *Computers and Mathematics*.
3. N. Tsao, "The Numerical Instability of Bini's Algorithm," *Information Processing Letters* 12, 1 (1980), 17–19.
4. N. Tsao, "Error complexity analysis of algorithms for matrix multiplication and matrix chain product," *IEEE Trans. on Computing* (in press).
5. W. Miller, "Computational Complexity and Numerical Stability," *SIAM J. on Computing* 4 (1975), 105–107.
6. V. Pan, "New Combination of Methods for the Acceleration of Matrix Multiplication," *Computers and Mathematics (with Applications)* 2 (1981), 73–125.
7. A. Aho, J. Hopcroft, J. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley (1974).
8. A. Borodin and I. Munro, *The Computational Complexity of Algebraic and Numeric Problems*, American Elsvier (1975).
9. E. Horowitz and S. Sahni, *Fundamentals of Computer Algorithms*, Computer Science Press (1978).
10. D. E. Knuth, *The Art of Computer Programming: Seminumerical Algorithms* 2, Addison-Wesley (1981).
11. D. Coppersmith and S. Winograd, "On the Asymptotic Complexity of Matrix Multiplication," Research Report, IBM T. J. Watson Research Center (December 1980).
12. S. Winograd, "Arithmetic Complexity of Computations," *SIAM* (1980).

LMED

8